(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－91435

(43)公開日　平成10年(1998)４月10日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　9/32	３５０	Ｇ０６Ｆ　9/32	３５０Ａ

審査請求　未請求　請求項の数６　ＯＬ　（全 15 頁）

(21)出願番号　特願平8－243375

(22)出願日　平成８年(1996)９月13日

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号

(72)発明者　大谷　寿賀子
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内

(72)発明者　岩田　俊一
東京都千代田区丸の内二丁目２番３号　三菱電機株式会社内

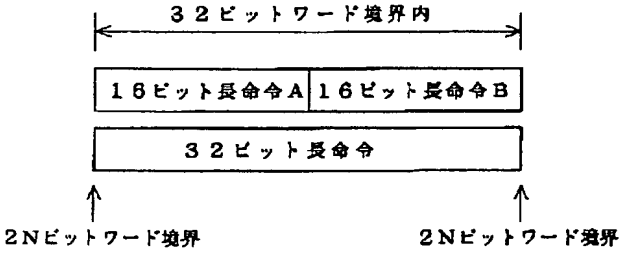
(74)代理人　弁理士　吉田　茂明　（外２名）

(54)【発明の名称】　二種類の命令長コードを実行するプロセッサ及びその命令コード入力装置

(57)【要約】

【課題】　二種類の命令長コードを備えたプロセッサにおいて、コードサイズの縮小化及びH／W量の削減化による高速化を図る。

【解決手段】　２種類の命令長（１６ビット，３２ビット）の命令コードを備えたプロセッサにおいて、命令コードの配置方法を次の２種類に制限する。即ち、（１）２個の１６ビット長命令コードを３２ビットワード境界内に格納し、（２）単一の３２ビット長命令コードはそのまま３２ビットワード境界内に格納する。更に、分岐先のアドレスを３２ビットワード境界に制限すると共に、各命令コードのＭＳＢに１ビットの命令長識別子を設けて命令コードの実行順序の制御を行う。これにより、プロセッサ内の命令フェッチ部から命令デコード部への転送経路は２種類となる。



【特許請求の範囲】
【請求項1】　N（Nは1以上の整数）ビット長命令を与える第1命令データ信号と2Nビット長命令を与える第2命令データ信号のみから成る命令コードを備え、前記第1及び第2命令データ信号の命令コードは、（1）2個の前記第1命令データ信号を2Nビット長のワードの境界内に格納し、（2）前記第2命令データ信号の各々を前記2Nビットワード境界内に格納するという規則の下に配置された、命令コード入力手段と、
前記命令コード入力手段をフェッチして前記規則の下に配置された前記命令コードを保持する命令フェッチ手段とを備えた、二種類の命令長コードを実行するプロセッサ。
【請求項2】　請求項1記載のプロセッサであって、
前記命令フェッチ手段より転送される前記命令コードを与える信号を受け取って、前記命令コードに対して分岐先アドレスの指定を前記2Nビットワード境界に制限することにより前記命令コードをデコードする命令デコード手段を更に備えた、二種類の命令長コードを実行するプロセッサ。
【請求項3】　請求項2記載のプロセッサであって、
前記第1及び第2命令データ信号のそれぞれは、その所定のビット位置において、命令実行順序の制御情報を与える命令長識別子データを備える、二種類の命令長コードを実行するプロセッサ。
【請求項4】　請求項2又は3記載のプロセッサであって、
外部割り込み信号及び特定のアドレス実行時に発生するPCブレーク割り込み信号のそれぞれによる割り込み要求を前記命令コードの前記2Nビットワード境界においてのみ受け付けて、前記割り込み要求に対応した処理を実行することを特徴とする、二種類の命令長コードを実行するプロセッサ。
【請求項5】　請求項2又は3記載のプロセッサであって、
前記2Nビットワード境界内に配置された前記命令コードが、ソフトウェアにより制御される割り込みの実行を指令するトラップ命令であるときには、前記命令デコード手段により前記トラップ命令をデコードして前記トラップ命令に対応した処理を実行することを特徴とする、二種類の命令長コードを実行するプロセッサ。
【請求項6】　プロセッサに対して命令コードを与える命令コード入力装置であって、
前記命令コードは、N（Nは1以上の整数）ビット長命令を与える第1命令データ信号と2Nビット長命令を与える第2命令データ信号のみから成り、
前記第1及び第2命令データ信号の命令コードを、
（1）2個の前記第1命令データ信号を2Nビット長のワードの境界内に格納し、（2）前記第2命令データ信号を前記2Nビットワード境界内に格納するという規則の下に配置する、命令コード入力装置。
【発明の詳細な説明】
【0001】
【発明の属する技術分野】この発明は、複数の命令コードの演算を実行するためのデータ処理装置に関するものであり、具体的には、プロセッサにおける命令コードの配置技術に関する。
【0002】
【従来の技術】通常、プロセッサにおいては、命令は、プロセッサとデータバスを介してつながっているメモリに、命令コードとして格納されている。この場合、当該メモリに格納されている命令コードのフォーマットには、①命令コードの長さが命令の種類によらずに常に一定に設定されている「固定長フォーマット」と、②命令コードの長さがそれぞれ命令の種類によって異なるように設定されている「任意長命令フォーマット」とがある。
【0003】命令コードには、演算、転送、分岐などの、命令の機能を指定するオペレーションコード部分と、命令の実行対象データ（オペランド）を指定するオペランドコード部分とがある。オペランドの指定は、命令コード中のアドレッシングモードの指定部において当該オペランドがレジスタ内に格納されているのか、それとも外部のメモリ内に格納されているのかを指定することにより、行われる。そして、オペランドがメモリ内にある場合には、更にアドレス情報を命令コード中に付加する。
【0004】以下に、固定長フォーマット及び任意長フォーマットの命令フォーマットを、それぞれ図18及び図19に模式的に示す。両図において、命令フォーマット100はオペランドがない場合を、命令フォーマット101はオペランドがある場合を、命令フォーマット102はオペランドがない場合の任意長フォーマットを、命令フォーマット103はオペランドがある場合の任意長フォーマットを、それぞれ表している。
【0005】
【発明が解決しようとする課題】
①命令コードが固定長命令フォーマットの場合
この場合には、命令コードのデコードが容易であるという利点がある。しかし、この命令フォーマットでは、決められた一定の命令長の範囲内で、オペレーションコードやアドレッシングモードの指定部や、さらにはアドレス情報などの付加情報を記述しなくてはならないという制約がある。従って、より多くの付加情報を記述するためには命令長を大きく設定する必要がある。その結果、命令長を大きくした固定長命令フォーマットでは、命令ビットパターンに冗長部分が増加し、コードサイズが大きくなるという問題点が生ずる。一方では、コードサイズを小さくするために命令長を小さく設定すると、命令機能に対する制限が大きくなるという問題点が生じる。

【０００６】②命令コードが任意長命令フォーマットの場合
この場合には、２種類以上の任意の命令長の命令フォーマットが使用されるので、各々の命令毎に応じて命令機能を拡張することができるという利点がある。また、オペランドのない命令の命令長を短く設定することができるので、固定長命令フォーマットの場合と比較して、コードサイズを小さくできる利点がある。
【０００７】その反面、メモリから読み込んだデータを各命令コードとして抽出し、更に、命令コード自身の各々をデコードする作業が複雑化するので、命令デコード方法が複雑にならざるを得ないという問題点がある。このため、メモリの内容から命令コードを抽出して、命令デコーダに送り込むためのH／W（ハードウェア）が大きくなる。例えば、図２０に示す様に、１６／３２ビット長命令フォーマット１０５，１０４を任意長命令フォーマットとして導入するときには、図２１に示す様に、命令フェッチ部と命令デコーダの間には、命令コードの転送のために、４つの経路を用意する必要がある。このためには、有効な命令コードをシフトして適切な命令コードの配置の下でデコードが実行されるように、複雑なシフト機能を命令デコーダに具備させなければならない。
【０００８】以上の通り、①の固定長命令フォーマット及び②の任意長命令フォーマットには、それぞれ一長一短がある。そこで、①及び②のそれぞれの利点を兼ね備えたプロセッサの実現が要望されているのである。
【０００９】この発明は、上記の懸案事項を実現するためになされたものであり、具体的には、（ｉ）固定長命令フォーマットに比べてコードサイズを縮小化し、且つ（ｉｉ）従来の任意長命令フォーマットに比べてプロセッサのH／W量を削減して高速化を図ることができる命令フォーマットを備えたプロセッサ及びプロセッサ用の入力装置を実現することを、その主たる目的としている。
【００１０】又、この発明は、命令実行中に発生する各割り込み（外部割り込み、ＰＣブレーク割り込み）やソフトウェア割り込みの機能を具備したプロセッサを実現することを、その副次的目的としている。
【００１１】又、この発明は、そのような命令コードをデコードするためのプロセッサの構成を具体化することを、その副次的目的としている。
【００１２】
【課題を解決するための手段】第１の発明に係る二種類の命令長コードを実行するプロセッサは、Ｎ（Ｎは１以上の整数）ビット長命令を与える第１命令データ信号と２Ｎビット長命令を与える第２命令データ信号のみから成る命令コードを備え、前記第１及び第２命令データ信号の命令コードは、（１）２個の前記第１命令データ信号を２Ｎビット長のワードの境界内に格納し、（２）前記第２命令データ信号の各々を前記２Ｎビットワード境界内に格納するという規則の下に配置された、命令コード入力手段と、前記命令コード入力手段をフェッチして前記規則の下に配置された前記命令コードを保持する命令フェッチ手段とを備えている。
【００１３】第２の発明に係る二種類の命令長コードを実行するプロセッサは、第１の発明の二種類の命令長コードを実行するプロセッサであって、前記命令フェッチ手段より転送される前記命令コードを与える信号を受け取って、前記命令コードに対して分岐先アドレスを前記２Ｎビットワード境界に制限することにより前記命令コードをデコードする命令デコード手段を更に備えている。
【００１４】第３の発明に係る二種類の命令長コードを実行するプロセッサであって、前記第１及び第２命令データ信号のそれぞれは、その所定のビット位置において、命令実行順序の制御情報を与える命令長識別子データを備えている。
【００１５】第４の発明に係る二種類の命令長コードを実行するプロセッサは、第２又は第３の発明の二種類の命令長コードを実行するプロセッサであって、外部割り込み信号及び特定のアドレス実行時に発生するＰＣブレーク割り込み信号のそれぞれによる割り込み要求を前記命令コードの前記２Ｎビットワード境界においてのみ受け付けて、前記割り込み要求に対応した処理を実行することを特徴としている。
【００１６】第５の発明に係る二種類の命令長コードを実行するプロセッサは、第２又は第３の発明の二種類の命令長コードを実行するプロセッサであって、前記２Ｎビットワード境界内に配置された前記第１命令データ信号の内の一方が、ソフトウェアにより制御される割り込みの実行を指令するトラップ命令であるときには、前記命令デコード手段により前記トラップ命令をデコードして前記トラップ命令に対応した処理を実行することを特徴としている。
【００１７】第６の発明に係る命令コード入力装置は、プロセッサに対して命令コードを与える命令コード入力装置であって、前記命令コードは、Ｎ（Ｎは１以上の整数）ビット長命令を与える第１命令データ信号と２Ｎビット長命令を与える第２命令データ信号のみから成り、前記第１及び第２命令データ信号の命令コードを、
（１）２個の前記第１命令データ信号を２Ｎビット長のワードの境界内に格納し、（２）前記第２命令データ信号の各々を前記２Ｎビットワード境界内に格納するという規則の下に配置している。
【００１８】
【発明の実施の形態】
（実施の形態１）ここでは、Ｎ＝１６とした場合の一例について、図面に基づき説明する。
【００１９】図１は、本発明に係る複数の演算を実行す

るデータ処理装置の構成を示すブロック図である。図１に示す通り、同装置は、プロセッサ１０を中核として構成される。

【００２０】プロセッサ１０は、演算部１、レジスタ２、プログラムカウンタ部（以下、ＰＣ部と称す）３、アドレス生成器４、命令キュー（ｑｕｅｕｅ）部５、命令フェッチ部６、命令デコード部７及び制御部８より構成されている。この内、制御部８は、命令デコード部７が出力するデコード結果に従って、プロセッサ１０内の各部１～７の動作を制御する。尚、図示の簡略化のために、同図中には、制御部８から出力される各制御信号の図示化は省略されている。又、演算部１は、ＡＬＵ（算術論理演算器）、シフタ（ｓｈｉｆｔ）、Ｌｏａｄ－Ｓｔｏｒｅユニット及び乗算器（Ｍｕｌ．）から成る。又、レジスタ２は汎用レジスタであり、３２ビット幅で１６本の信号線を有し、演算結果のデータ及びアドレスデータを保持する。他の構成部分の詳細については、後述する。

【００２１】尚、命令キュー部５と命令フェッチ部６とを総称して、「命令フェッチ機能部」と定義する。

【００２２】又、プロセッサ１０と、外部の周辺回路１１、メモリ１２及びデータセレクタ１４とは、データバス１５とバスインタフェイス部１３とによって接続されている。尚、データバス１５は、命令コードを与える命令コードデータ信号を本プロセッサ１０に入力する「入力手段」に該当する。

【００２３】Ｉ．命令コードの配置方法（命令セット）

次に、この発明の根幹をなす「命令コードの配置方法」について説明する。

【００２４】既述した通り、❶固定長命令フォーマット及び❷任意長命令フォーマットでは、それぞれに一長一短があった。特に、❷の任意長命令フォーマットにおいて、１６ビット長命令コードと３２ビット長命令コードの２種類の命令コードを用いる場合は、命令フェッチ部（ないし命令キュー部）と命令デコード部間には、４つの転送経路が必要とされる。この内、特にハードウェアの複雑化という深刻な問題点を惹起せしめる要因となっているのは、３２（２Ｎ）ビット長命令が３２（２Ｎ）ビット境界を乗り越えてしまうような命令コードの配置を許していることにあると考えられる。この点を、図２に模式的に示す。

【００２５】今、命令フェッチ部が外部のメモリより同図の上段側に示す様な順序で配列された４つの命令コード１０６～１０９をフェッチしてきたものとし、これらの内で、３２ビット長の命令コード１０７及び１０８がデータ処理にとって有効であるものとする。このような場合には、同図の下段側に示すような配置として、有効命令コード１０７，１０８をデコードする必要がある。従って、同図に示す様な、交差した２つの経路１１０，１１１を実現する必要が生じるのである。このような経路となる命令コードの配置を、ここでは「２Ｎビット命令が２Ｎビット境界をまたぐ配置」と呼称している。このために、命令デコーダ側のハードウェアが複雑化せざるを得ないのである。かかる要因に着眼するならば、そのような命令コードの配置を禁止する必要がある。

【００２６】そこで、この発明では、（ｉ）Ｎ（Ｎ≧１）ビット長の命令（第１命令データ信号）と２Ｎビット長の命令（第２命令データ信号）という、２種類の命令長のみから成る複数の命令コード（命令コードデータ信号）を実行することとし、（ｉｉ）命令コードの配置方法、即ち、２Ｎビット長で与えられる「ワード」内（これを、２Ｎビット長ワード境界内と称す）に各命令をどのように配置するのかという点を、以下の２種類のみに制限することとしている。即ち、

（１）２個のＮビット長命令を２Ｎビットワード境界内に格納する。

【００２７】（２）単一の２Ｎビット長命令を２Ｎビットワード境界内に格納する。

【００２８】本実施例では、Ｎ＝１６の場合を扱うものとしているので、１６ビット長命令と３２ビット長命令（最大ビット長の命令）の２種類のみとなり、命令コードの配置方法は図３に示す通りとなる。

【００２９】このような配置制限を設けることによって、３２ビット長命令が３２ビット境界をまたぐような命令コードの配置が完全に禁止されることとなり、その結果として、図１の命令フェッチ部６と命令デコード部７との転送経路は、図４に示す様に、３種類に削減される。この３種類の転送経路は、次の模式的な説明図５によって、理解されうるであろう。

【００３０】同図に示す様に、命令キュー部５内に保持されている状態では、３種類の命令コードの配列がありうる。その内、同図の上段及び中段に示されるものは、それぞれ、先行する１６ビット長命令コードＡ１及び後行する１６ビット長命令コードＢ２のみが有効となる場合であり、いずれも上記配列方法（１）の制限の下で配列されている。この内、同図の上段の場合には、図４に示す経路ＲＴ１を経て、命令コードＡ１はデコーダに転送される。又、図５の中段の場合には、図４に示す交差経路ＲＴ３を経て、命令コードＢ２はデコーダに転送される。他方、図５の下段に示される３２ビット長の命令コードＣ１は上記配置方法（２）に基づく制約の下で配列されているものであり、図４に示す経路ＲＴ１，ＲＴ２を経てデコーダに転送される。従って、デコード化のためには、３種類の経路ＲＴ１～ＲＴ３のみで良いこととなる。

【００３１】このように、命令コードの配置を上記（１），（２）の制約に服させることとしているので、転送経路を従来の場合よりも１種類分だけ削減することが可能となるのであるが、上記（１），（２）の制約を受けて配置された命令コードを与える命令コードデータ

信号は、図１のデータバス１５上の入力データ信号として実現されて、命令キュー部５に保持される。そのためには、ここでは、上記制約ルール（１），（２）に基づいたプログラム制御によって、命令コードのデータをメモリ１２内に書き込んでいる。従って、上記制約ルール（１），（２）に基づく配置で書き込まれた命令コードデータ信号を記憶する「メモリ１２」と、上記配置順序に従って当該メモリ１２から読み出された命令コードデータ信号をプロセッサ１０内に入力する手段たる「データバス１５」とを、「プロセッサ用命令コード入力装置」として総称する。

【００３２】尚、上記命令コードデータ信号の内、上記制約（１）に基づき配置された１６ビット（一般的にはＮビット）長命令コードを与えるものを「第１命令コードデータ信号」と、上記制約（２）に基づき配置された３２ビット（一般的には２Ｎビット）長命令コードを与えるものを「第２命令コードデータ信号」と、それぞれ呼称する。

【００３３】プロセッサ１０の命令キュー部５内の命令コードデータ信号のフォーマットを、図６に示す。同図中の記号ｏｐ１，ｏｐ２はオペランドコードを、記号Ｒ１，Ｒ２はレジスタを、記号Ｃは定数を、記号ｃｏｎｄは分岐条件の指定を表わす。

【００３４】尚、上記制限（１），（２）の下でメモリ１２に命令コードのデータを書き込むのに代えて、上記制限とは無関係に命令コードのデータをメモリ１２内に書き込み、新たにデータバス１５上にメモリ１２より読み出した命令コードデータ信号を上記制限（１），（２）の下で配置する機能部を設け、この機能部の出力データを命令キュー部５に格納するようにしても良い。

【００３５】ＩＩ．分岐先の制御

更に、本プロセッサ１０においては、分岐先のアドレスを３２ビット境界にのみ指定ないし制限する。このように、命令コードの３２ビット境界への配置を禁止したことに加えて、分岐先アドレスを３２ビット境界にのみに制限したことにより、命令フェッチ部６と命令デコーダ間の転送経路は２種類にまで削減される。

【００３６】この点は、既述した図５に立ち戻って概観すれば、容易に理解されうるであろう。即ち、分岐先アドレスを３２ビット境界においてのみ指定できるように制限したということは、図１の命令デコード部７では、命令フェッチ部６より出力された、３２ビット長ワード境界内に配置された命令コードデータ信号を受け取った後は、当該命令コードデータ信号をその３２ビット（２Ｎビット）境界からデコードを開始することを意味する。従って、３２ビットワード境界内に２個の１６ビット長命令コードが配置されている場合は、先行する命令Ａ２をデコードした後に、同図の中段の命令コードＢ２を、先行する命令コードＡ２があるビット位置へシフトした上でデコードすればよい。

【００３７】よって、命令キュー部５から命令デコード部７への命令コードデータ信号の転送経路は、図７に模式的に示す様に、２種類のみとなるのである。即ち、転送経路は、

（イ）命令キュー部：上位１６ビット→命令デコード部：上位１６ビット。

【００３８】（ロ）命令キュー部：下位１６ビット→命令デコード部：下位１６ビット。

【００３９】分岐命令の飛び先の指定は、以下の形式で行われる。その点を、図８、図９及び図１０に示す本プロセッサ１０における命令コード表に基づき説明する。尚、上記命令コード表において、「Ｆｏｒｍａｔ」中の「ｄｅｓｔ」は結果収納先のレジスタの番号を示しており、「ｓｒｃ」は演算対象であり、ここでは図６に示したレジスタＲ１を意味しており、当該レジスタＲ１中の数値がメモリのアドレス値となっている。又、「ｐｃｄｉｓｐ８」は、即値が８ビットで与えられていることを示している。

【００４０】上記命令コード表において、

（ａ）ＪＭＰ，ＪＬ命令は、命令コード内で指定したレジスタの値が分岐先アドレスとなる。（ただし、レジスタの下位２ビットの値”０”は無視する）。

（ｂ）ＢＲＡ，ＢＬ，ＢＣ，ＢＮＣ命令は、８ビット又は２４ビットの即値を指定する。

（ｃ）ＢＥＱ，ＢＮＥ，ＢＥＱＺ，ＢＮＥＺ，ＢＬＴＺ，ＢＧＥＺ，ＢＬＥＺ，ＢＧＴＺ命令は１６ビットの即値を指定する。

上記（ｂ），（ｃ）における分岐先のアドレスは、

（分岐命令のＰＣ値）＋（符号拡張された即値を左に（最上位ビット位置側に）２ビットシフトした値）

とする。ただし、加算を行う際には、ＰＣ値の下位２ビットは”００”となる。

【００４１】以上のように、分岐先を３２ビット境界のみに指定したことにより、分岐先アドレスの下位２ビットは常に”００”となる。したがって、命令コード内で分岐先アドレスを指定する場合には、その下位２ビットを指定する必要がなくなる。結果として、命令コード内から直接分岐できる範囲は2^2倍、従って４倍となる。本実施の形態の場合には、上記命令コード表に示されるように、命令コード内のアドレス指定部は最大で２４ビットであるので、実行中の命令のアドレスから±３２ＭＢｙｔｅの範囲に直接分岐できる。

【００４２】ＩＩＩ．命令のデコード順序の制御

更に本プロセッサ１０においては、命令コード内に命令フォーマット識別子を与える情報を所定のビット数として、ここでは１ビットとして設けている。そして、この命令フォーマット識別子の値いかんによって、命令のデコード及び実行順序を制御している。ここでは、各命令コードのＭＳＢ（Ｍｏｓｔ　Ｓｉｇｎｉｆｉｃａｎｔ　Ｂｉｔ）が命令フォーマット識別子に該当する。

【００４３】上記命令フォーマット識別子を用いた制御のルールは、次の通りに設定される。即ち、単一の３２ビット長命令コードのＭＳＢは常に１に設定されている。他方、命令コードが２個の１６ビット長命令コードから成る場合においては、上位１６ビット側に存在する命令コードのＭＳＢは常に０とされる。それに続く下位１６ビット側の命令コードに対しては、そのＭＳＢの値いかんによって異なる処理を行う。

【００４４】命令の実行順序の制御方法を、図１１を用いて説明する。同図において、①「命令Ｂ」のＭＳＢが０の場合は、「命令Ａ」と「命令Ｂ」とは連続して実行される。

【００４５】それに対して、②「命令Ｂ」のＭＳＢが１の場合は、「命令Ａ」のみが実行される。命令Ｂは実行されない。即ち、３２ビット長命令の命令コード配置の制約に基づいたワードアライメント調整のための命令Ｂとしてワードアライメント調整用「ＮＯＰ命令」を挿入したときには、アセンブラが自動的に当該「ＮＯＰ命令」にあたる命令コードのＭＳＢを１とし、これにより「命令Ａ」のみの実行が行われる。通常の無効演算「ＮＯＰ命令」は”０１１１０００００００００００”として与えられるが、アライメントという上記目的のために「ＮＯＰ命令」は３２ビットワード境界内の下位側１６ビット位置に挿入されたのであって、それを直接実行する必要もない。従って、本プロセッサ１０では、「ＮＯＰ命令」は”１１１１０００００００００００”として与えられ、その結果、「ＮＯＰ命令」自体は実行されない。

【００４６】このような命令実行順序の制御を行うことにより、コード配置を満たすために挿入された無効演算である「ＮＯＰ命令」の実行時間ペナルティがなくなるという利点が得られる。

【００４７】Ｖ．命令デコード方法とその制御部

以下では、命令のデコード制御方法の具体例を、図１２を用いて説明する。

【００４８】（命令デコード部７の構成）命令デコード部７は、命令デコード入力ラッチ７ａ、命令デコーダ７ｃ、定数生成器７ｄ及び命令デコード部７の制御ロジック７ｂから構成される。これらの内で、制御ロジック７ｂは、命令デコード部７の各部の制御を司る。又、命令デコーダ７ｃは、命令デコード入力ラッチ７ａに格納された３２ビットのビットパターンのうちの有効な命令コードを、入力として受け取り、当該命令デコーダ７ｃは命令コードをデコードする。尚、命令デコード入力ラッチ７ａ内の命令コードの配置は、メモリ１２上での命令コードの配置と同じである。デコード結果はプロセッサ１０の制御部８へと出力され、制御部８は、そのデコード結果に基づき、演算部１やプロセッサ１０全体を制御する。

【００４９】各部のより詳細な動作は、次の通りである。

【００５０】（命令フェッチ部６）先ず、命令フェッチ部６は、データバス１５（図１）を介してメモリ１２から命令コードデータ信号を３２ビット長単位でフェッチし、それらを命令キュー部５に格納する。更に命令フェッチ部６は、後述の第５制御信号ＣＢに応じて、命令キュー部５より順次に命令コードデータ信号を読み出して命令デコード部７へ転送する。その結果、転送されてきた命令コードデータ信号は、３２ビット幅の命令デコード入力ラッチ７ａへ格納される。

【００５１】命令デコード入力ラッチ７ａに格納された命令コードデータ信号の内で有効な命令コードを与える信号のデコードの実行は、既述した命令フォーマット識別子に基づき、制御ロジック７ｂにより、次の通りに制御される。その点を、以下に詳述する。

【００５２】（制御ロジック７ｂ）図１２の第１～第３制御信号ＣＴ１～ＣＴ３（入力信号とも称す）と命令デコード入力ラッチ７ａ内の有効コード配置との関係を、図１３に示す。図中の斜線部は命令フォーマット識別子を示し、梨地部は有効なコードを示す。

【００５３】図１２に示すように、制御ロジック７ｂは、命令デコード入力ラッチ７ａに納められた３２ビットパターンのうちの命令長識別子を与える第１，第２制御信号ＣＴ１，ＣＴ２と、制御ロジック７ｂ自身から出力される第３制御信号ＣＴ３とを、入力とする。即ち、命令デコード入力ラッチ７ａは、命令フォーマット識別子として自ら保有する３２ビットのビットパターンの０ビット目の値と１６ビット目の値とを、各々第１制御信号ＣＴ１及び第２制御信号ＣＴ２として、制御ロジック７ｂに出力する。第３制御信号ＣＴ３は制御ロジック７ｂ自身からの出力信号であり、それは、命令デコーダ７ｃが出力するデコード終了信号ＥＳに応じて、上記ラッチ７ａに納められた３２ビットのビットパターン中の０ビット目から１５ビット目までの部分がデコードされているのか（この場合には”０”）、それとも１６ビット目から３１ビット目までの部分が現在デコードされているのか（この場合には”１”）を、表わす。

【００５４】（α）第１制御信号ＣＴ１の値が”１”である時は、命令デコード入力ラッチ７ａに納められている命令コードデータ信号は３２ビット長命令を与えるものである。

【００５５】（β）第１制御信号ＣＴ１が”０”であるときは、命令デコード入力ラッチ７ａには２個の１６ビット長命令から成る命令コードデータ信号が格納されている。このとき、さらに第２制御信号ＣＴ２が”０”かつ第３制御信号ＣＴ３が”０”であれば、有効な命令コードは上位１６ビット側にある。これに対して、第２制御信号２が”０”かつ第３制御信号ＣＴ３が”１”であれば、有効な命令コードは下位１６ビット側にある。

【００５６】（γ）第１制御信号ＣＴ１が”０”で、し

かも第２制御信号ＣＴ２が”１”である場合は、上位１６ビットの命令コードのみが有効なコードであり、下位１６ビットの命令コードは、上述したワードアライメント調整用「ＮＯＰ命令」であり実行されない。

【００５７】制御ロジック７ｂは、以上の入力信号ＣＴ１～ＣＴ３の値を基に、次コードのデコードのために、次の３種類の制御信号ＣＡ，ＣＢ，ＣＴ３を出力する。この内、第４制御信号ＣＡは、命令デコード入力ラッチ７ａ内において下位１６ビットの命令コードを上位１６ビット位置へシフトするための制御信号、即ちシフタ制御信号であり、第５制御信号ＣＢは命令アドレスの３２ビット境界のポインタであり、命令フェッチ部６に対して命令デコード入力ラッチ７ａへの命令コードの転送開始を命令する信号である。第３制御信号ＣＴ３は、既述した通り、次の有効コードが３２ビットワード境界内の上位１６ビット位置にあるのか、下位１６ビット位置にあるのかを示す信号である。

【００５８】さらに、図１２に示すように、制御ロジック７ｂは、第６制御信号ＣＣとして、命令デコーダ７ｃにおいてデコード実行中の命令コードのアドレスが３２ビットワード境界上にある場合には”１”を、ワード境界上にない場合には”０”を、後述するＰＣのビット３０部（ＰＣ３０部）に出力する。

【００５９】命令コード内の即値により分岐先アドレスを指定する分岐命令、即ち、上述した（ｂ）ＢＲＡ命令等や（ｃ）ＢＥＱ命令等をデコードする場合は、命令デコード入力ラッチ７ａの後半２４ビット（８ビット目から３１ビット目まで）位置に接続された定数生成器７ｄへ、同ラッチ７ａは即値を出力する。定数生成器７ｄは、即値を符号拡張して得られる値を、バスＳ２に出力する。

【００６０】命令実行時に分岐が発生した場合には、第１～第３制御信号（入力）ＣＴ１～ＣＴ３の値に関わりなく、命令デコード部７の第４制御信号ＣＡ、第５制御信号ＣＢ、第６制御信号ＣＣ及び第３制御信号ＣＴ３は、すべて初期化される。

【００６１】制御信号（入力）と制御信号（出力）の関係を図１４に示す。

【００６２】以上より、命令デコード部７（図１，図１２）においては、いずれの場合にも、第１回目にデコードする命令コードは、必ず命令デコード入力ラッチ７ａの先頭からはじまる。したがって、命令デコード入力ラッチ７ａの内容が、そのまま命令デコーダ７ｃに転送される。命令コードデータ信号が単一の３２ビット長命令コードの場合には、転送経路は図１２に示す経路Ｐ２，Ｐ３である。又、上位の１６ビット長命令コードの転送経路は、経路Ｐ２である。

【００６３】命令デコード入力ラッチ７ａにおける命令コードの配置が１６ビット長命令の逐次実行である場合のみ、命令コードの２回目の命令デコーダ７ｃへの出力が行われる。この際、２回目の命令デコーダ７ｃへの出力前に、次のような処理が行われる。デコード対象となる命令コード、即ち有効なコードは３２ビットワード境界内の下位側の１６ビット位置にあるので、命令デコード入力ラッチ７ａの内容は左へ（３２ビットのビットパターンの０ビット目方向へ）１６ビット分だけシフトされ（経路Ｐ１）、入力ラッチ７ａに入る。その結果が、２回目の命令デコーダ７ｃへの出力となる（経路Ｐ２）。

【００６４】このように、経路Ｐ１を経由するのは、有効コードが命令デコード入力ラッチ７ａの下位１６ビット位置にある場合に限られるので、Ｈ／Ｗ量を削減することができ、さらに高速化を図れることが可能となる。

【００６５】ＶＩ．ＰＣ部３のインクリメント動作

図１５は、図１中のＰＣ部３及びアドレス生成器４の部分を拡大したブロック図である。図１５を用いてＰＣ部３のインクリメント動作について説明する。

【００６６】アドレス生成器４は、シフタ４ａと加算器４ｂとから構成されており、分岐命令のアドレスをアドレッシングモードに従って計算する。

【００６７】ＰＣ部３は、プログラムカウンタ（以後、ＰＣと称す）を中核として、更に比較器３ｂ、（＋１／０）部３ｅ、バックアップ・プログラムカウンタ（以下、ＢＰＣと称す）３ｆ、ＢＰＣ３０部３ｇ及びプログラムカウンタ・ブレークポインタ（以下、ＰＢＰと称す）より構成される。この内、ＰＢＰ３ａとＢＰＣ３ｆとは、制御レジスタである。

【００６８】上述のプログラムカウンタは３２ビットのカウンタであり、現在実行中の命令のアドレス値を保持する。命令コードの配置方法を上述の通り限定しているので、本プロセッサ１０（図１）の命令は偶数アドレス値からのみ始まる。よって、プログラムカウンタの３１ビット目の値は、図２２に例示するように、”０”固定となる。従って、ハードウェア上では、ＰＣの３１ビット目の値を実現する必要がないので、ＰＣは、図１５に示す通り、０ビット目から２９ビット目までの値を与えるＰＣ（０：２９）３ｃと、３０ビット目の値を与えるＰＣ３０部３ｄとによって実現されている。この内、ＰＣ３０部３ｄは、第６制御信号ＣＣの値”０”又は”１”を保有するレジスタである。

【００６９】又、上述のＢＰＣは、後述する割り込み・トラップ発生時にＰＣ３ｃが保持するＰＣ値を退避する。ここで、図２３に例示するように、ＢＰＣの３１ビット目の値は常に”０”固定されているので、ハードウェア上では、ＢＰＣは、０ビット目から２９ビット目までの値を与えるＢＰＣ（０：２９）３ｆと、３０ビット目の値を与えるＢＰＣ３０より実現されている。従って、本プロセッサ１０（図１）は、後述する割り込み・トラップの発生を検出すると、ＰＣ（０：２９）３ｃの値をＢＰＣ（０：２９）３ｆへ退避させる。他方、ＰＢ

Ｐ３ａは、後述するＰＣブレーク割り込みを制御するための３２ビット幅制御レジスタであり、ＰＢＰ３ａは、割り込みを起動する命令実行のアドレス値を予め保有しており、制御部８（図１）から出力するバスＤ１上の書き込み命令信号を用いて、上記アドレス値を書き込んでいる。

【００７０】ＰＣ（０：２９）３ｃの更新は、本プロセッサ１０（図１）において分岐以外の命令が実行された時には、以下に示すように行われる。尚、ここでは、４本のバスＳ１，Ｓ２，Ｓ３及びＤ１を使用しており、信号線Ｄは、ＰＣ３ｃに接続されている。

【００７１】図１２の命令デコード部７の制御ロジック７ｂから出力される第５制御信号ＣＢ（３２ビットワード境界のアドレスを指すポインタ）が更新されると、ＰＣ（０：２９）３ｃが保持する値は（＋１／０）部３ｅによって”＋１”だけ増加させられ、その増加後の値が信号線Ｄ上にのる。これに対して、第５制御信号ＣＢが更新されない場合は、ＰＣ（０：２９）３ｃの値は、（＋１／０）部３ｅによって増加されることなく、そのまま信号線Ｄ上にのる。

【００７２】信号線Ｄ上の値はＰＣ（０：２９）３ｃへ書き込まれ、これにより、ＰＣ（０：２９）３ｃのＰＣ値は、次命令のＰＣ（０：２９）３ｃの値へと書き換えられる。また、信号線ＤはバスＳ１に結線されているので、このＰＣ（０：２９）３ｃの更新値をバスＳ１上に呼びだすことができる。

【００７３】ＰＣ３０部３ｄには、命令デコード部７内の制御ロジック７ｂ（図１）から出力される第６制御信号ＣＣの値が、命令実行時に書き込まれる。

【００７４】他方、命令実行時に分岐が発生した場合には、次命令のＰＣ値は、アドレス生成器４を用いて生成され、その生成値はバスＳ３を経由してＰＣ（０：２９）３ｃに書き込まれる。

【００７５】又、飛び先アドレス指定を即値でおこなう分岐命令（上述した分岐命令（ｂ），（ｃ））の場合には、命令デコード部７の定数生成器７ｄ（図１）は、当該命令コードから抽出した即値を３２ビットに拡張し、拡張後の即値をバスＳ２を介してアドレス生成器４へ出力する。そして、符号拡張された即値は、アドレス生成器４内のシフタ４ａによって左に（最上位側へ）２ビットだけシフトされる。ＰＣ（０：２９）３ｃの更新値は、信号線Ｄからバス Ｓ１を介して、アドレス生成器４に出力される。ここで、ＰＣ（０：２９）３ｃの更新値は、分岐命令のＰＣ上位３０ビットにあたる。加算器４ｂは、

（ＰＣ（０：２９）３ｃの更新値）＋（符号拡張された即値を左に２ビットシフトした値）

を計算し、これにより、分岐先アドレスの上位３０ビットが得られる。この上位３０ビットの値は、バスＳ３を介して、ＰＣ（０：２９）３ｃに書き込まれる。

【００７６】飛び先アドレス指定をレジスタ２（図１）の値で行う分岐命令の場合（上述の分岐命令（ａ））には、命令コードにより指定されたレジスタ２から、バスＳ１を介して、レジスタに保持されている値が受け取られ、その定数値がＰＣ３ｃに書き込まれる。

【００７７】分岐発生時は、命令デコード部７が第６制御信号ＣＣを”０”に初期化することにより、ＰＣ３０部３ｄには”０”が書き込まれる。よって、本プロセッサ１０（図１）では、上述の通り、分岐先のアドレスは３２ビットワード境界のみとなる。

【００７８】ＶＩＩ．割り込み・トラップ動作の説明

プロセッサ１０（図１）が通常のプログラムを実行している途中で、ある事象が発生すると、そのプログラムの実行を中断して、別のプログラムを実行する必要が生じる。このような事象としては、大別して、割り込み及びトラップ動作がある。

【００７９】（ａ）割り込み（Ｉｎｔｅｒｒｕｐｔ）

上記事象の内、外部からのハードウェア信号（これを割り込み要求信号と称す）により、又は特定のアドレス実行時に生ずるＰＣブレーク信号により発生する事象。

【００８０】（ｂ）トラップ（Ｔｒａｐ）

上記事象のうち、命令により発行される事象。

【００８１】本プロセッサ１０（図１）はまた、外部割り込み（ＥＩ）及びＰＣブレーク割り込み（ＰＢＩ）から成る２種類の（ａ）割り込みと、１種類のトラップ（ＴＲＡＰ）とを実現する機能を備える。

【００８２】割り込み・トラップ発生時の処理手順を図１６を用いて説明すれば、概要、次の通りとなる。

【００８３】割り込み要求信号又はＰＣブレーク信号が有効となった場合には、後程詳述する様に、プロセッサ１０は、３２ビットワード境界でのみ割込み命令を受け付ける。また、トラップ命令は、その命令実行後、トラップの処理を開始する。

【００８４】これにより、プロセッサ１０（図１）は、プログラムの実行を中断して割り込み又はトラップの処理を行う。その際、同プロセッサ１０は、後述する様に、割り込み又はトラップの事象を検出して図１５のＰＣ３ｃのＰＣ値をＢＰＣ３ｆへ退避させることとしており、その後、各々の割り込み又はトラップに対応した処理プログラム「割り込み・トラップ処理ハンドラ」への分岐をおこなう。

【００８５】「割り込み・トラップ処理ハンドラ」での処理が完了すると、「割り込み・トラップ処理ハンドラ」からの復帰命令を実行し、続いてＰＣ３ｃのＰＣ値の復帰を行い、プロセッサ１０は当該割り込み・トラップ処理から復帰する。

【００８６】本プロセッサ１０における割り込み・トラップの処理は、以上の通り、ハードウェアが処理を行う部分と、プログラムが処理をする部分とから成る。即ち、本プロセッサ１０では、上述の処理の内、（１）戻

り先であるＰＣ値のＢＰＣ３ｆへの退避、（２）「割り込み・トラップ処理ハンドラ」への分岐、（３）ＢＰＣ値のＰＣ３ｃへの書き込み、を、ハードウェア部分が実行する。

【００８７】前述の外部割り込み（ＥＩ）は、外部からのハードウェア信号（割込み要求信号）により発生する。割込み要求信号による割り込みの要求は、３２ビットワード境界上でのみ受け付ける（この機構は、後記の検出回路の構成による）。割り込み発生時に図１５のＢＰＣ３ｆに退避される値は、次命令のＰＣ値である。

【００８８】他方、ＰＣブレーク割り込み（ＰＢＩ）は、特定のアドレスを実行したときに発生する。本プロセッサ１０（図１）においては、指定するアドレスは３２ビットワード境界のみである。各サイクル毎に、図１５の比較器３ｂは、ＰＢＰ３ａが保持する値とＰＣ３ｃの値とを比較し、両者の値が一致するときにＰＣブレーク信号１８を出力する。ＰＣブレーク信号１８は、後述する割り込み・トラップ検出回路を介して、割り込み・トラップ検出信号として検出され、その信号は図１５の（＋１／０）部３ｅへ出力される。その結果、プロセッサ１０に割り込みが発生し、ＰＣ値の退避が生じる。既述した通り、ＰＢＰ３ａへの書き込みは、データバスＤ１を用いて行われる。又、割り込み発生時にＢＰＣ３ｆに退避される値は、次命令のＰＣ値である。

【００８９】又、トラップとは、ソフトウェアで制御する割り込みのことであり、トラップ命令の実行により発生する。この場合、トラップ命令が３２ビットワード境界内の上位１６ビット側にあるのか、それとも下位１６ビット側にあるのかを与える情報を、ＢＰＣビット３０、即ち、ＢＰＣ３０部３ｇに格納する。トラップ命令が上位１６ビットにある場合は、ＢＰＣ３０部３ｇの値は”０”、下位１６ビットである場合は、ＢＰＣ３０部３ｇの値は”１”である。割り込み発生時にＢＰＣに退避される値は、（トラップ命令のＰＣ値＋４）である。

【００９０】割り込み・トラップを検出する回路を、図１７に示す。同検出回路は、図１の制御部８の一部を構成しており、インバータ２２、ＡＮＤ回路２３，２４及び割り込み・トラップ検出回路１６より成る。

【００９１】本プロセッサ１０は、当該検出回路によって、外部信号１９による割り込みの要求及びＰＣブレーク信号１８による割り込みの要求を、共に３２ビットワード境界においてのみ受け付ける。即ち、３２ビットワード境界においては第６制御信号ＣＣのレベルは”０”であるので、第６制御信号ＣＣを反転したワード境界検出信号２１（そのレベルは”１”）と割込み要求信号１９とが同時に有効（”１”）になったときにのみ、出力信号ＶＥが有効（”１”）となる。また、ワード境界検出信号２１とＰＣブレーク信号１８とが同時に有効（”１”）になったときにのみ、出力信号ＶＦが有効となる。このように、３２ビットワード境界でのみ上記割込み要求が受け付けられ、割り込み・トラップ処理ハンドラへ分岐し、当該処理を行った後に、「割り込み・トラップ処理ハンドラ」からの復帰命令の実行によって、割り込み・トラップ処理から実行プログラムへの復帰が行われる。

【００９２】他方、トラップ命令では、トラップ命令による割り込みが指令されると、トラップ要求信号２０が有効（”１”）になる。

【００９３】トラップ要求信号２０、出力信号ＶＥ、出力信号ＶＦの内のいずれか１つが有効（”１”）になると、割り込み・トラップ検出回路１６により、割り込み・トラップ検出信号１７が出力される。割り込み・トラップ検出信号１７が出力されると、割り込み・トラップ処理ハンドラへ分岐し、当該処理を行った後、復帰命令を実行し、当該処理から実行プログラムへと復帰する。

【００９４】既述の通り、割り込み・トラップ発生時には、ＰＣ３ｃ（図１５）の値がＢＰＣ３ｆに退避される。割り込み発生時、ＢＰＣ３ｆには次命令のＰＣ値が書き込まれる。

【００９５】例えば、分岐命令の実行直後に割り込みが発生した場合には、図１５のアドレス生成器４で生成された分岐先を与えるＰＣ値は、制御部８の制御の下、バスＳ３を経由してＢＰＣ３ｆに書き込まれる。これに対して、分岐命令以外の命令の実行直後に割り込みが発生した場合には、（＋１／０）部３ｅによってＰＣ（０：２９）３ｃの出力値に”＋１”を加えられたものが、ＢＰＣ（０：２９）３ｆに書き込まれる。

【００９６】ＢＰＣ３０部３ｇには、ＰＣ３０部３ｄの値が書き込まれる。本プロセッサ１０（図１）においては、既述した通り、割り込みは３２ビットワード境界においてのみ受け付けられるので、割り込み発生時にＢＰＣ３０部３ｇに退避される値は、常に”０”である。

【００９７】トラップ発生時には、（（トラップ命令のＰＣ）＋４）の値がＢＰＣ３ｆに書き込まれる。即ち、（＋１／０）部３ｅによってＰＣ（０：２９）３ｃの出力値に”＋１”が加えられたものが、ＢＰＣ（０：２９）３ｆに書き込まれ、ＢＰＣ３０部３ｇにはＰＣ３０部３ｄの値が書き込まれる。即ち、トラップを発生させたトラップ命令が３２ビットワード境界の上位１６ビットにある場合は、ＢＰＣ３０部３ｆには”０”が、３２ビットワード境界の下位１６ビットにある場合はＢＰＣ３０部３ｇには”１”が、それぞれ書き込まれる。

【００９８】割り込み・トラップからの復帰は、既述の通り、リターン命令を実行することによって行う。リターン命令は、制御部８が出力する信号２５を受けてＢＰＣ３ｆが出力する信号が与えるアドレスへ分岐する。ただし、本プロセッサ１０においては、分岐先のアドレスは常に３２ビット境界のみに設定されているので、ＢＰＣ３ｆがＰＣ３ｃに復帰する際、ＰＣの下位２ビットは常に”００”となる。

【0099】（まとめ）以上の構成を採用したことにより、次の様な特徴点が得られる。

【0100】本プロセッサ10は、Nビット長と2Nビット長という2種類の命令長から成る命令コードを実行することにより、固定長命令フォーマットに比べて、命令機能の制限を受けることなくコードサイズを縮小することができると共に、従来の任意長命令フォーマットに比べて命令デコード方法を簡略化できる。

【0101】特に、命令コードの配置に一定の制限を加えて、2Nビット境界をまたぐ命令の配置を禁止すると共に、分岐先アドレスを2Nビットワード境界のみに指定することとしたことにより、命令フェッチ部6及び命令デコード部7間のデータ転送経路を格段に削減することができる。その結果、命令デコードのためのH／W量を削減でき、高速化が図れる。

【0102】しかも、本プロセッサ10においては、分岐先アドレスを2Nビット境界に制約したことにより、分岐先アドレスの下位2ビットは常に”00”となる。したがって、命令コード内で分岐先アドレスの下位2ビットを指定する必要がない。その結果として、アドレスの全てのビットを指定する場合と比較して、2^2倍、即ち4倍の広範囲へ実行中の命令のアドレスから直接分岐することができる。

【0103】そのような機能を備えた本プロセッサ10に対して、更に、各種の割り込みやトラップ処理をサポートし得る機能を備えることも可能である。

【0104】

【発明の効果】請求項1～5記載の各発明によれば、命令コード入力手段により2個の第1命令データ信号及び第2命令データ信号は共に2Nビットワード境界内に格納されるので、2Nビット長命令のデータが当該2Nビット境界をまたいでしまうこととなる命令コードの配置が禁止されることとなる。このため、命令フェッチ部から命令デコード部への命令コードデータ信号の転送経路を従来の4種類から3種類に削減することができるという効果がある。

【0105】特に、請求項2記載の発明によれば、命令デコーダ手段が分岐先アドレスを2Nビットワード境界内に制限するので、命令コードデータ信号の転送経路を2種類までに削減することができるという効果がある。

【0106】更に、請求項3記載の発明によれば、命令長識別子に基づいて各命令のデコード及びその実行順序を制御することが可能となる。そして、命令長識別子の適切な設置によっては、無効演算としての命令コードデータ信号を実行しなくて済むようにすることができ、これにより実行時間ペナルティをなくすことも可能になるいう利点がある。

【0107】更に、請求項4記載の発明では、外部割り込み及びPCブレーク割り込みをサポートすることができるという効果がある。

【0108】又、請求項5記載の発明では、トラップ命令をサポートしうるという効果も得られる。

【0109】又、請求項6記載の発明では、命令コード入力装置が2個の第1命令データ信号及び第2命令データ信号を共に2Nビットワード境界内に格納した上で、そのように各命令コードの配置が制約された2Nビット長の命令コードデータ信号を順次にプロセッサに対して入力させることができるので、当該命令コードデータ信号を受けるプロセッサの側においては、命令デコード化のための転送経路を従来の4種類から3種類に削減できるという効果を、プロセッサに対してもたらし得るという効果がある。

【図面の簡単な説明】

【図1】　データ処理装置の構成を示すブロック図である。

【図2】　命令コードの配置方法を模式的に示す図である。

【図3】　命令コードの配置方法を模式的に示す図である。

【図4】　命令フェッチ部と命令デコード部との転送経路を模式的に示す図である。

【図5】　3種類の命令コードの配列を示す図である。

【図6】　命令キュー部内の命令コードデータ信号のフォーマットを示す図である。

【図7】　命令キュー部と命令デコード部との転送経路を模式的に示す図である。

【図8】　プロセッサの命令コード表を示す図である。

【図9】　プロセッサの命令コード表を示す図である。

【図10】　プロセッサの命令コード表を示す図である。

【図11】　命令コードの実行順序の制御方法を示す図である。

【図12】　命令デコード部の構成の詳細を示すブロック図である。

【図13】　第1～第3制御信号と命令デコード入力ラッチ内の有効コード配置との関係を示す図である。

【図14】　命令デコード部の制御ロジックの第1～第3制御信号と第4～第6制御信号との関係を示す図である。

【図15】　PC部及びアドレス生成器の構成の詳細を示すブロック図である。

【図16】　割り込み・トラップ発生時の処理手順を示す図である。

【図17】　割り込み・トラップを検出する回路を示すブロック図である。

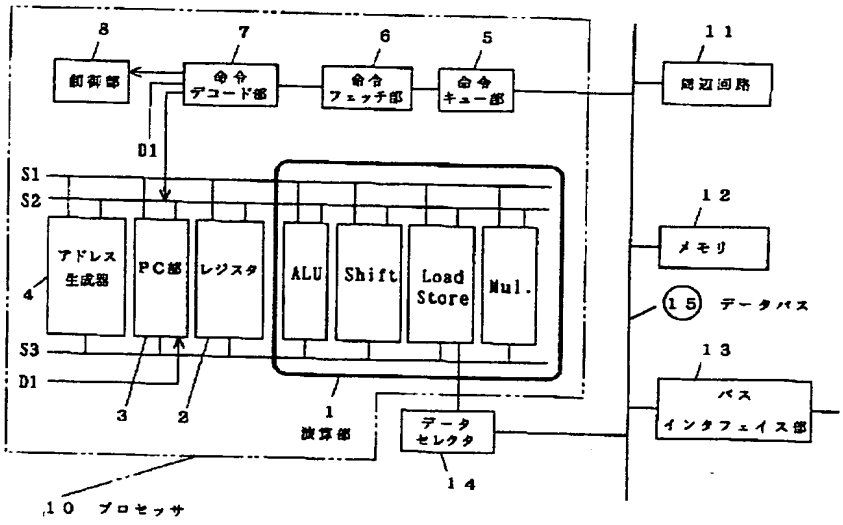
【図18】　従来の固定長命令フォーマットを示す図である。

【図19】　従来の任意長命令フォーマットを示す図である。

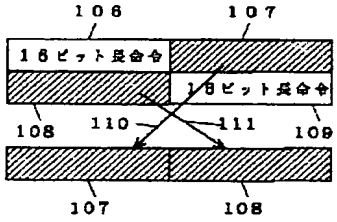
【図20】　従来の任意長命令フォーマットの具体例を

示す図である。
【図21】　従来の命令フェッチ部と命令デコーダ部間の転送経路を示す図である。
【図22】　PCがハードウェア上で実現されている態様を示す図である。
【図23】　BPCがハードウェア上で実現されている態様を示す図である。
【符号の説明】
1　演算部、2　レジスタ、3　PC部、3a　プログラムカウンタ・ブレークポインタ、3b　比較器、3c　プログラムカウンタ部（ビット0～29）、3d　プログラムカウンタ部（ビット30）、3e　（÷1／0）部、3f　バックアップ・プログラムカウンタ（ビット0～29）、3g　バックアップ・プログラムカウンタ（ビット30）、4　アドレス生成器、5　命令キュー部、6命令フェッチ部、7　命令デコード部、7a　命令デコード入力ラッチ、7b制御ロジック、7c　命令デコーダ、7d　定数生成器、8　制御部、10プロセッサ、11　周辺回路、12　メモリ、13　バスインタフェイス部、14　データセレクタ、15　データバス、16　割り込み・トラップ検出回路、17　割り込み・トラップ検出信号、18　PCブレーク信号、19　割込み要求信号、20　TRAP要求信号、21　ワード境界検出信号、22　インバータ、23，24　AND回路、25　出力信号。

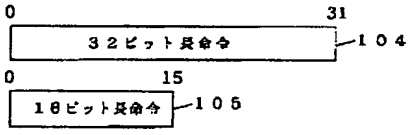
【図1】



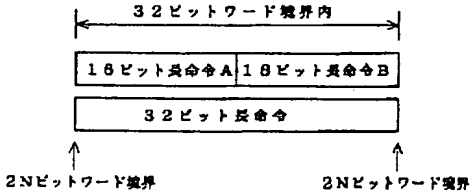
【図2】



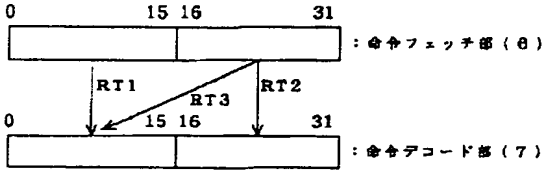
【図20】



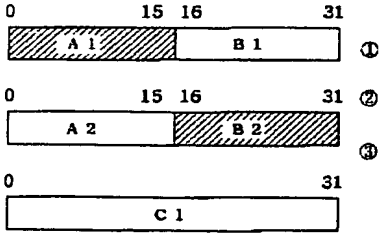
【図3】



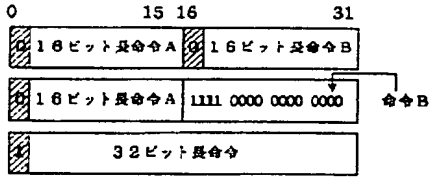
【図4】



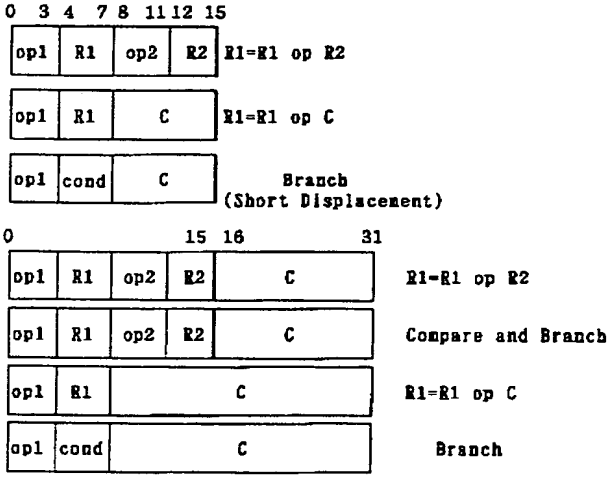
【図5】



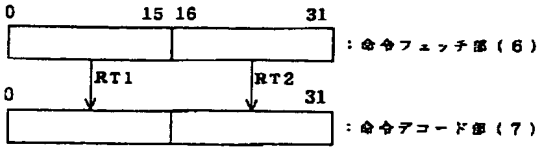
【図11】



【図6】



【図7】

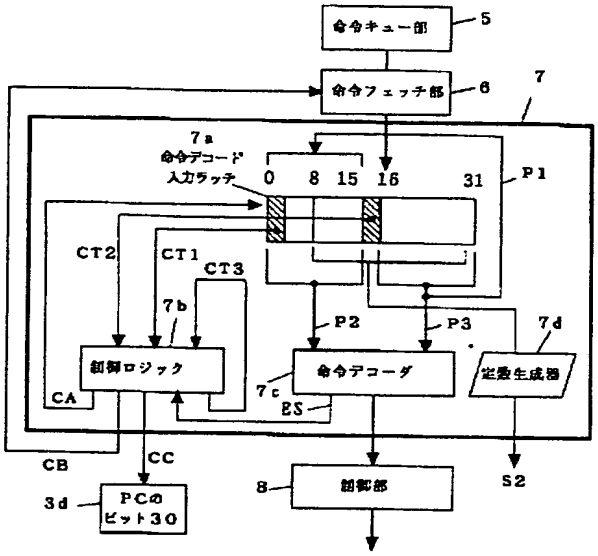


【図8】

mnemonic	Function	Format
LD	Load	0010 dest 1100 src
		0010 dest 1110 src
		1010 dest 1100 src disp16
LD	Load byte	0010 dest 1000 src
		1010 dest 1000 src disp16
LDUB	Load unsigned byte	0010 dest 1001 src
		1010 dest 1001 src disp16
LDH	Load halfword	0010 dest 1010 src
		1010 dest 1010 src disp16
LDUH	Load unsigned halfword	0010 dest 1011 src
		1010 dest 1011 src disp16
LOCK	Load locked	0010 dest 1101 src
ST	Store	0010 src1 0100 src2
		0010 src1 0110 src2
		0010 src1 0111 src2
		1010 src1 0100 src2 disp16
STB	Store byte	0010 src1 0000 src2
		1010 src1 0000 src2
STH	Store halfword	0010 src1 0010 src2
		0010 src1 0010 src2 disp16
UNLOCK	Store unlocked	0010 src1 0101 src
LD24	Load 24-bit immediate	1110 dest imm24
LDI	Load immediate	0110 dest imm8
		1001 dest 1111 0000 imm16
MV	Move register	0001 dest 1000 src
MVFC	Move from control register	0001 dest 1001 src
MVTC	Move to control register	0001 dest 1010 src
SETH	Set high-order 16-bit	1101 dest 1100 0000 imm16
CMP	Compare	0000 src1 0100 src2
CMPI	Compare immediate	1000 0000 0100 src imm16
CMPU	Compare unsigned	0000 src2 0101 src2
CMPUI	Compare Unsigned immediate	1000 0000 0101 src imm16
ADD	Add	0000 dest 1010 src
ADD3	Add 3-operand	1000 dest 1010 src imm16
ADDI	Add immediate	0100 dest imm8
ADDV	Add(with overflow checking)	0000 dest 1000 src
ADDV3	Add 3-operand(with overflow checking)	1000 dest 1000 src imm16
ADDX	Add with carry	0000 dest 1001 src
NEG	Negate	0000 dest 0011 src
SUB	Subtract	0000 dest 0010 src
SUBV	Subtract(with overflow checking)	0000 dest 0000 src
SUBX	Subtract with borrow	0000 dest 0001 src

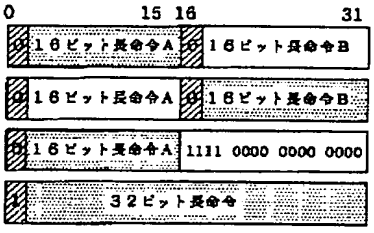
BL1

【図12】

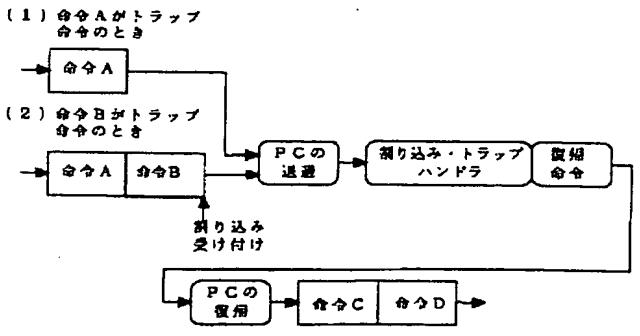


【図13】

第1制御信号 CT1	第2制御信号 CT2	第3制御信号 CT3
0	0	0
0	0	1
0	1	*
1	*	*



【図16】



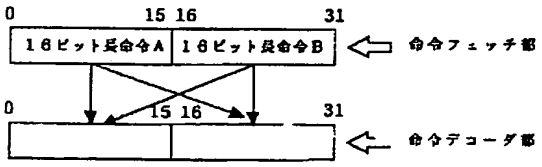
【図9】

BL1

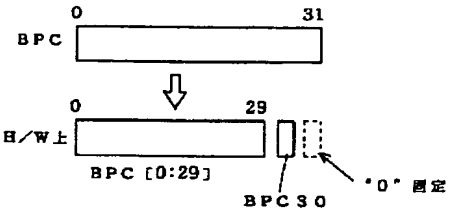
AND	AND	0000 dest 1100 src	
AND3	AND 3-operand	1000 dest 1100 src	imm16
NOT	Logical NOT	0000 dest 1011 src	
OR	OR	0000 dest 1110 src	
OR3	OR 3-operand	1000 dest 1110 src	imm16
XOR	Exclusive OR	0000 dest 1101 src	
XOR3	Exclusive OR 3-operand	1000 dest 1101 src	imm16
DIV	Divide	1001 dest 0000 src 0000 0000 0000 0000	
DIVU	Divide unsigned	1001 dest 0001 src 0000 0000 0000 0000	
MUL	Multiply	0001 dest 0110 src	
REM	Remainder	1001 dest 0010 src 0000 0000 0000 0000	
REMU	Remainder unsigned	1001 dest 0011 src 0000 0000 0000 0000	
SLL3	Shift left logical 3-operand	1001 dest 1100 src	imm16
SLLI	Shift left logical Immediate	0101 dest010 imm5	
SRA	Shift right arithmetic	0001 dest 0010 src	
SRA3	Shift right arithmetic 3-operand	1001 dest 1010 src	imm16
SRAI	Shift right arithmetic Immediate	0101 dest 001 imm5	
SRL	Shift right logical	0001 dest 0000 src	
SRL3	Shift right logical 3-operand	1001 dest 1000 src	imm16
SRLI	Shift right logical Immediate	0101 dest 000 imm5	
BC	Branch on C-bit	0111 1100 pcdisp8	
		1111 1100 pcdisp24	
BEQ	Branch on equal	1011 src1 0000 src2	pcdisp16
BEQZ	Branch on equal zero	1011 0000 1000 src	pcdisp16
BGEZ	Branch on greater than or equal zero	1011 0000 1011 src	pcdisp16
BGTZ	Branch on greater than zero	1011 0000 1101 src	pcdisp16
BL	Branch and link	0111 1110 pcdisp8	
		1111 1110 pcdisp24	

BL2

【図21】



【図23】



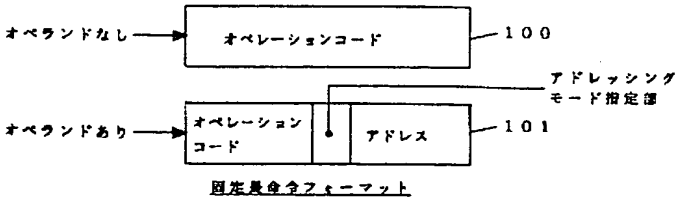
【図10】

BL2

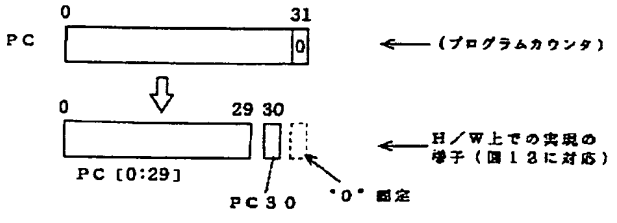
BLEZ	Branch on less than or equal zero	1011 0000 1100 src	pcdisp16
BLTZ	Branch on less than zero	1011 0000 1010 src	pcdisp16
BNC	Branch on not C-bit	0111 1101 pcdisp8	
		1111 1101 pcdisp24	
BNE	Branch on not equal	1011 src1 0001 src2	pcdisp16
BNEZ	Branch on not equal zero	1011 0000 1001 src	pcdisp16
BRA	Branch	0111 1111 pcdisp8	
		1111 1111 pcdisp24	
JL	Jump and link	0001 1110 1100 src	
JMP	Jump	0001 1111 1100 src	
NOP	No operation	0111 0000 0000 0000	
RTE	Return from Interruput,Trap	0001 0000 1101 0110	
TRAP	Trap	0001 0000 1111 imm4	

MACHI	Multiply-accumulate high-order halfwords	0011 src1 0100 src2
MACLO	Multiply-accumulate low-order halfwords	0011 src1 0101 src2
MACWHI	Multiply-accumulate word and high-order halfwords	0011 src1 0110 src2
MACWLO	Multiply-accumulate word and low-order halfwords	0011 src1 0111 src2
MULHI	Multiply high-order halfwords	0011 src1 0000 src2
MULLO	Multiply low-order halfwords	0011 src1 0001 src2
MULWHI	Multiply word and high-order halfwords	0011 src1 0010 src2
MULWLO	Multiply word and low-order halfwords	0011 src1 0010 src2
MVFACHI	Move from accumulator high-order word	0101 src 1111 0000
MVFACLO	Move from accumulator low-order word	0101 src 1111 0001
MVFACMI	Move from accumulator middle-order word	0101 src 1111 0010
MVTACHI	Move to accumulator high-order word	0101 src 0111 0000
MVTACLO	Move to accumulator low-order word	0101 src 0111 0001
RAC	Round accumulator	0101 0000 1001 0000
RACH	Round accumulator halfword	0101 0000 1000 0000

【図18】



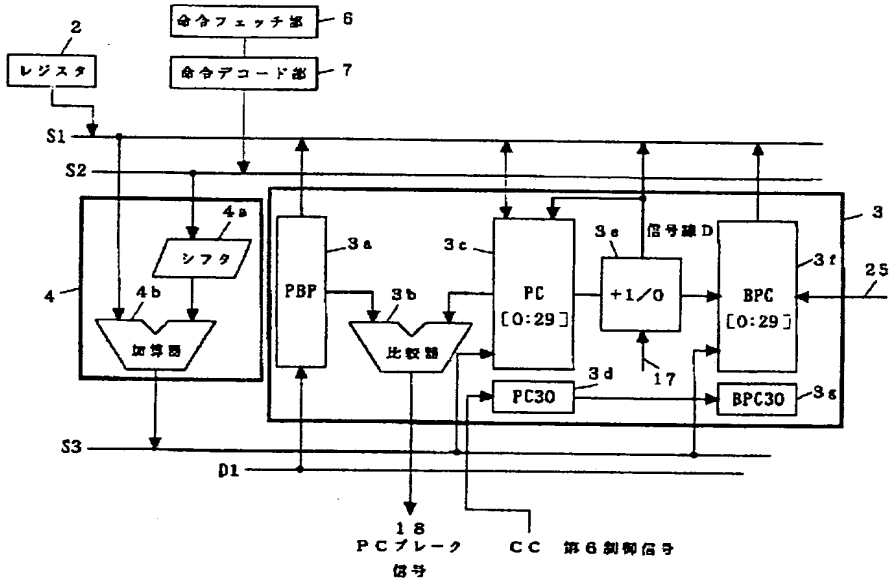
【図22】



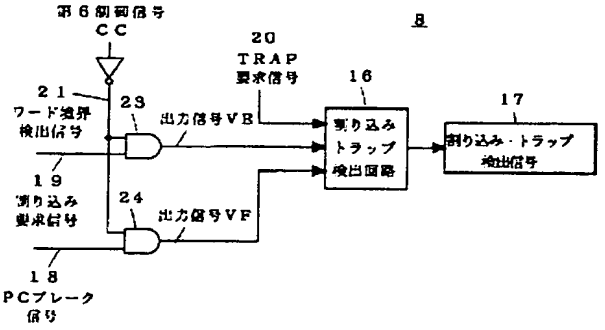
【図14】

分岐発生	入力			シフタ制御	32ビット境界線ポインタ	次コードの場所	PCのbit30
	CT1	CT2	CT3	CA	CB	CT3	CC
なし	0	0	0	シフトする	更新する	下位16bit	0
	0	0	1	シフトしない	更新しない	上位16bit	1
	0	1	*	シフトしない	更新しない	上位16bit	0
	1	*	*	シフトしない	更新する	上位16bit	0
あり	*	*	*	シフトしない	初期化	0	0

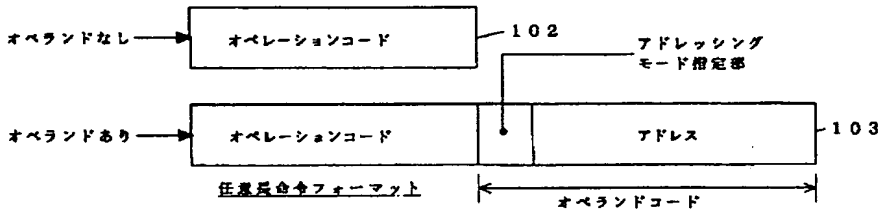
【図15】



【図17】



【図19】


オペランドなし
オペレーションコード
102
アドレッシング
モード指定部
オペランドあり
オペレーションコード
アドレス
103
任意長命令フォーマット
オペランドコード